

Para saber más: el directorio WEB-INF

Tomcat y el mundo Servlet ya tienen una forma estándar de trabajar con login, contraseña, permisos y recursos protegidos. Todo esto se puede configurar a través del archivo **web.xml** y una pequeña configuración en el servidor Tomcat.

La idea es que la aplicación web defina qué debe tener un login, cuáles son los permisos y recursos (URLs) protegidos. Todo esto está en **web.xml**. Aquí hay un ejemplo de un **web.xml** que protege la URL `/entrada` y las páginas `.html` :

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

    <display-name>gerenciador</display-name>

    <welcome-file-list>
        <welcome-file>bienvenido.html</welcome-file>
    </welcome-file-list>

    <login-config>
        <auth-method>BASIC</auth-method>
    </login-config>

    <security-role>
        <role-name>ADMINISTRADOR</role-name>
    </security-role>

    <security-role>
        <role-name>USUARIO</role-name>
    </security-role>
```

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>acceso controlador</web-resource-name>
    <url-pattern>/entrada</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>ADMINISTRADOR</role-name>
  </auth-constraint>
</security-constraint>

<security-constraint>
  <web-resource-collection>
    <web-resource-name>acceso a páginas html</web-resource-name>
    <url-pattern>*.html</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>USUARIO</role-name>
  </auth-constraint>
</security-constraint>
</web-app>
```

[COPIA EL CÓDIGO](#)

En este ejemplo, tenemos dos permisos (ROLES): ADMINISTRADOR y USUARIO . Para acceder a /entrada es necesario ser ADMINISTRADOR , y para acceder a *.html ser USUARIO .

Por otro lado, Tomcat se encarga de cargar los usuarios y permisos. Una forma sencilla de hacerlo es utilizar el archivo **tomcat-users.xml** dentro del proyecto Servers , en la carpeta Tomcat. Simplemente reemplázelo con el contenido a continuación, para definir dos usuarios (admin y user):

```
<tomcat-users>
  <role rolename="ADMINISTRADOR"/>
  <role rolename="USUARIO"/>
  <user username="admin" password="123" roles="ADMINISTRADOR, I
  <user username="user" password="123" roles="USUARIO"/>
</tomcat-users>
```

[COPIA EL CÓDIGO](#)

Observe que el `admin` tiene los permisos de `ADMINISTRADOR` y `USUARIO`, el `user` solo `USUARIO`.

Todo esto fué definido dentro de otro estándar, llamado Java Authentication and Authorization Service (**JAAS - API estándar de Java para seguridad**), sin embargo, no se usa tanto en las aplicaciones web de Java.